

INFORMATICA - CdL in FISICA

PROVA SCRITTA DEL 27/06/2013

Scrivere **in stampatello** COGNOME, NOME e MATRICOLA su ogni foglio consegnato

N.B.: In tutti gli esercizi viene valutata anche la leggibilità del codice proposto. Inoltre, non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione (come, ad esempio, `continue`, `break` e istruzioni di `return` all'interno di cicli che ne provochino l'uscita forzata).

Utilizzare il tipo boolean definito da `typedef enum {false, true} boolean` invece di interi come variabili booleane;
In tutti gli esercizi non è consentito l'uso di variabili `static`

ESERCIZIO 1 (6 punti)

Si dice **sequenza di multipli** una sequenza di interi positivi tali che ogni intero (tranne il primo) sia un multiplo del precedente. La sequenza termina quando l'utente immette un intero che viola la sequenza, cioè che è negativo o è positivo ma non multiplo del precedente. Il carattere che termina la sequenza non è considerato appartenere alla sequenza. Si legga una sequenza di multipli e si stampi la posizione del secondo intero pari (se esiste) e la lunghezza totale della sequenza.

Se ad esempio la sequenza digitata dall'utente è:

```
"-1"
```

un possibile output del programma è:

```
Mi dispiace non ci sono stati 2 numeri pari.  
Lunghezza seq. : 0
```

mentre se la sequenza digitata dall'utente è:

```
"3 4"
```

un possibile output del programma è:

```
Mi dispiace non ci sono stati 2 numeri pari.  
Lunghezza seq. : 1
```

Se la sequenza digitata dall'utente è:

```
"3 15 60 62"
```

un possibile output del programma è:

```
Mi dispiace non ci sono stati 2 numeri pari.  
Lunghezza seq. : 3
```

Infine se la sequenza digitata dall'utente è:

```
"3 15 60 180 -1"
```

un possibile output del programma è:

```
Pos secondo pari:4  
Lunghezza seq. : 4
```

continua

ESERCIZIO 2 (5 punti)

Si definisca in C un funzione ricorsiva con un parametro A array di tipo array di interi che calcola true se l'array contiene almeno un elemento che verifica la seguente condizione: $A[i] = A[i+1] - A[i-1]$ e restituisce nel parametro i (passato per indirizzo) la posizione del primo elemento in A che verifica tale condizione.

ESERCIZIO 3 (6 punti)

Scrivere una funzione che controlli che un array di interi costituisca una *piramide (Maya) rovesciata*. Si dice che un array e' una piramide (Maya) rovesciata se e' costituito da una sequenza non vuota di interi ordinati in maniera strettamente decrescente, seguita da una sequenza eventualmente vuota di interi uguali, seguita infine da una sequenza non vuota di interi ordinati in maniera strettamente crescente.

ESERCIZIO 4 (14 punti)

Si vogliono rappresentare gli ordini di un fornitore di ciliege. Ogni ordine contiene il codice del cliente che ha fatto l'ordine, il quantitativo espresso in kg di ciliege prenotate e un campo che dice se il cliente e' stato fin a quel momento in regola con i pagamenti degli ordini precedenti. Gli ordini sono collezionati in una lista nella quale gli ordini dei clienti in regola con i pagamenti precedono quelli dei clienti in ritardo sui pagamenti. A parita' di situazione dei pagamenti poi gli ordini sono ordinati in base al codice cliente.

- (i) **(1 punti)** Si definiscano i tipi di dato necessari per implementare in C la rappresentazione indicata. Si identifichi con `Ordini` il tipo di dato principale.

Si definiscano le seguenti operazioni su oggetti di tipo `Ordini` mediante opportune procedure o funzioni.

- (ii) **(5 punti)** Dato una lista di ordini, e un quantitativo (espresso in kg) di ciliege disponibili, scrivere una funzione *ricorsiva* che evada, cancellandoli, tutti gli ordini che possono essere evasi per l'intero quantitativo richiesto rispettando l'ordine della lista. La funzione deve restituire il numero di kg di ciliege che avanzano.

Ad esempio se i quantitativi corrispondenti agli ordini della lista sono

$5- > 7- > 4- > 6- > 2$

e abbiamo 10 kg a disposizione la funzione deve restituire 1 e modificare la lista ottenendo

$7- > 6- > 2;$

- (iv) **(4 punti)** Data una lista di ordini, un codice cliente, un quantitativo ordinato e una situazione di pagamento, scrivere una procedura che inserisca il nuovo ordine nella lista;
- (v) **(4 punti)** Data una lista di ordini, scrivere una funzione che controlli che la quantita' di ciliege ordinate dai clienti in regola con i pagamenti costituisca almeno i due terzi della quantita' totale di ciliege ordinate.

N.B. Le procedure/funzioni **NON** devono contenere alcuna istruzione di input/output (ad es. `scanf`, `printf`, `getchar`, `putchar`, ...)